Index

Symbols

3DES ciphers
  reviewing 21

A

Abstract Syntax Notation One (ASN.1) 6, 103, 139

Advanced Encryption Standard (AES) 5, 17, 61
  cipher, reviewing 20, 21
  decrypting with, on command line 37-41
  encrypting with, on command line 37-41

Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) 9

Advanced Encryption Standard in Galois/Counter Mode with a Synthetic Initialization Vector (AES-GCM-SIV) mode 32
  reviewing 32

Advanced Encryption Standard New Instructions (AES-NIs) 21

AES programmatically
  used, for decrypting openssl enc file 49, 50
  used, for encrypting openssl enc file 44, 45

Application Programming Interface (API) 4, 37, 78, 174
  bindings 7
  compatibility 11
  instability 11

Application-Specific Integration Circuits (ASICs) 86

ARIA algorithm 23

ARIA cipher 24

ARM 4

ARMv8.2-SHA 62

ARMv8 Cryptographic Extension 21, 61, 62

asymmetric cryptography algorithms 4, 120

asymmetric encryption algorithms 4, 16, 96, 120
  need for 96
  private key 96
  public key 96

authenticated encryption (AE) 30, 76, 164

Authenticated Encryption with Associated Data (AEAD) 30, 166

authentication tags 71

avalanche effect 56

B

base CRL 281
Basic Input/Output (BIO) 254
bit rotations 17
BLAKE2 64
BLAKE2b 64, 66
BLAKE2 family
reviewing, of hash functions 64, 65
BLAKE2s 64, 66
BLAKE3 65
blockchain 58
block cipher
padding for 33
versus stream cipher 17, 18
block cipher operation modes 17, 25, 32
AES-GCM-SIV mode, reviewing 32
Cipher Block Chaining (CBC) mode, reviewing 27, 28
Counter (CTR) mode, reviewing 28, 29
Electronic Code Book (ECB) mode, reviewing 25, 26
Galois/Counter Mode (GCM) mode, reviewing 30-32
selecting 32
block counter 23
blocking sockets 253
Blowfish cipher 23
BoringSSL 10
OpenSSL, comparing with 10, 11
Botan 7
OpenSSL, comparing with 7, 8
Browser Exploit Against SSL/TLS (BEAST) 22, 169
brute force 19
brute-force resistant 85
BSD systems 9

C

C 7
C++ 7
C++11 7
C++17 7
Camellia algorithm 23
Camellia cipher 24
Carter-Wegman + CTR (CWC) mode 32
CAST5 cipher 23
CAST-128 23
Central Processing Unit (CPU) 19
certificate
generating, for web and email client 294-296
generating, for web server 292-294
revoking 296, 297
Certificate Authority (CA) 125, 141, 166, 172, 202, 244
certificate chain 140, 141
certificate depth 195
certificate pinning 244
Certificate Revocation List (CRL) 140, 202, 203, 245, 281
generating 297-300
Certificate Revocation List (CRL), using in C programs 203-205
CRL lookup callback, implementing 206, 207
CRL lookup callback, registering 205
function, implementing for downloading CRL from distribution point 208, 209
function, implementing for downloading CRL from HTTP URL 210, 211
program, running 211, 212
certificate revocation status
providing, via OCSP 301-304
certificate signing chain 140-144

Certificate Signing Request (CSR) 144, 145, 172, 245, 283
Certificate Status Request 213
Certificate Transparency (CT) 280
certificate verification chain 140
ChaCha20 8
ChaCha20 cipher 22
reviewing 22
ChaCha-Poly1305 9
ChaCha stream cipher 65
chain of trust 140
chosen-plaintext attack 28
cipher 16
Cipher Block Chaining (CBC) mode
reviewing 27, 28
cipher feedback (CFB) mode 32
ciphertext 16
client certificate
generating 294-296
packaging, into PKCS #12 container files 226-228
collision 56
collision attacks 23, 59
command line
HMAC, calculating 77, 78
leaf certificate, verifying 154
message digest, calculating 67
OCSP, using on 214-217
symmetric encryption key, deriving from password 87, 88
TLS client connection, establishing on 170-172
TLS server connection. accepting on 173, 174
Common Name (CN) 179
Compression Ratio Info-Leak Made Easy (CRIME) 169
Context (CTX) 47
Continuous Integration 43
Counter (CTR) mode
reviewing 28, 29
Counter with CBC-MAC (CCM) mode 32
C programs
custom verification, of peer certificates 194-196
OCSP, using in 218, 219
CRL number 281
CRYPTO_BUFFER functionality 10
cryptocurrencies 58
cryptographic algorithm 4, 56
cryptographically secure pseudo random generator (CSPRNG) 35
Cryptographic Doom Principle 76
cryptographic hash functions 56, 66
MDC-2 66
properties 56
RIPEMD-160 66
security, assessing 59, 60
selecting 66
Whirlpool 66
cryptography 3
custom verification of peer certificates, in C programs 194-196
program, running 200-202
verification callback, implementing 198-200
verification callback, registering 197

D

Data Encryption Standard (DES) 21
ciphers, reviewing 21
Datagram Transport Layer Security (DTLS) 165
data integrity verification 56, 57
delta CRL 281

denial-of-service 59

deterministic 85

Diffie-Hellman (DH) key exchange method 29, 88, 98, 164

digest program

- implementing 68, 69
- running 69

Digital Signature Algorithm (DSA) 63, 98

- reviewing 122
- selecting 125

Digital Signature Algorithm (DSA), supported by OpenSSL

- reviewing 122
- ECDSA, reviewing 123
- EdDSA, reviewing 124, 125
- overview 121
- RSA, reviewing 121, 122
- SMA, reviewing 125

digital signatures 4, 57, 72, 120

- ec-sign program, implementing 130
- ec-verify program, implementing 134
- features 120
- using programmatically 129, 130
- verifying programmatically 133
- versus MACs 121

Distinguished Encoding Rules (DER) 103, 139, 202, 300

Distinguished Name (DN) format 139

DNS poisoning 140

Domain Validated (DV) certificates 145

DTLS protocol 4

dynamic linking 6

E

ec-sign program

- implementing 130-132
- running 132, 133

ec-verify program

- implementing 134, 135
- running 135

Electronic Code Book (ECB) mode 25

- reviewing 25, 26

ElGamal algorithm 99

Elliptic Curve Cryptography (ECC) 101

Elliptic Curve Diffie-Hellman (ECDH) 98

Elliptic Curve Digital Signature Algorithm (ECDSA) 98

- reviewing 123, 124

elliptic curve keypair

- generating 126, 127

Encrypt-and-MAC (E&M) scheme 75

encryption key 16, 75, 76

encryption modes 25

encrypt-then-authenticate-then-translate (EAX) mode 32

Encrypt-then-MAC (EtM) scheme 75

ENGINE API 5

engines 5

Envelope API 46

- disadvantages 46

Envelope (EVP) 46

Ephemeral Diffie-Hellman (DHE) 164

Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) 164

Ethereum 63

EtM TLS protocol extension 76

EVP API 68

exhaustive search 19

existential forgery attack 72

existential forgery under a chosen-message attack 73

Extendable Output Functions (XOFs) 62

Extended Validation (EV) certificates 145, 146

extra data 197

F

Field-Programmable Gate Arrays (FPGAs) 86

file descriptor (fd) 174

filter BIOs 174

free and open source software (FOSS) projects 6

G

Galois/Counter Mode (GCM) mode

reviewing 30-32

Git 58

GMP 6

GNU ecosystem 6

GnuPG 63

GNU Privacy Guard (GnuPG) 23

GNU Project 6

GnuTLS 6

OpenSSL, comparing with 6, 7

GOST12 65

GOST89 65

GOST89 cipher 23, 24

GOST94 65

GOST2012 65

GOST2015 cipher 24

GOvernment STandard (GOST) 24

Graphical Processing Units (GPUs) 86

H

handshake secret 167

handshaking operation 99

Hard Core Library (HCL) 7

hash-and-sign paradigm 120

Hash-based Message Authentication Code (HMAC) 57, 73-75, 166, 225

calculating, on command line 77, 78

calculating, programmatically 78

Hash-based Message Authentication Code (HMAC) program

implementing 79, 80

running 81

hash collision 56

HashEdDSA 124

HMAC-based key derivation function (HKDF) 167

HMAC function 73

HMAC-SHA-256 function 74

hybrid encryption scheme 99

Hybrid Public Key Encryption (HPKE) 99

Hypertext Transfer Protocol (HTTP) 165

I

IDEA cipher 23

Individual Validation (IV) certificates 146

initialization vector (IV) 167

Input Key Material (IKM) 84

Input/Output (I/O) 253

Integrated Encryption Scheme (IES) 99

Intel SHA extensions 61, 62

intermediate CA certificates 141

generating 287-291

significance, reasons 142

intermediate CA config file

reference link 290

International Data Encryption Algorithm (IDEA) 25

Internet Engineering Task Force (IETF) 169

Internet-of-Things (IoT) devices 8, 170

IPsec 77

irreversible 85

K

KangarooTwelve (K12) 62
kdf program
implementing 89, 90
running 90
KECCAK Message Authentication Code (KMAC) 74
Kernel TLS (KTLS) 5
key agreement 164
Key Derivation Function (KDF) 84
overview, supported by OpenSSL 87
parameters 85
key exchange 4, 76, 164
operation 99
keypair 96
key signing parties 97
key signing party 97
keystream 18
Kuznyechik 24

L

leaf certificate 141
verifying, on command line 154
verifying, programmatically 154, 155
x509-verify program, implementing 155-158
x509-verify program, running 158
length extension attack 74
LibreSSL 9
OpenSSL, comparing with 9
Libtasn1 6
lightweight TLS libraries 8
MatrixSSL 8
Mbed TLS 8
OpenSSL, comparing with 8, 9
wolfSSL 8
Local Area Networks (LANs) 24

M

MAC function 72
MAC function security 72, 73
MAC-then-Encrypt (MtE) scheme 75
Magma 24
Man in the Middle (MITM) attack 32, 96, 97, 140, 164, 244
key fingerprint, verifying over phone 97
key, signing by trusted third party 98
key splitting 98
meeting, in person 97
master secret 167
MatrixSSL 8
Mbed TLS 8
MD4 64
MD5 61, 64, 65
MD family
functions 64
reviewing, of hash functions 64
Mercurial 58
Merkle tree structure 65
message authentication 72
Message Authentication Code (MAC) 22, 57, 72, 164
versus digital signatures 121
Message Digest (MD) 4, 56, 64
blockchain 58
calculating, on command line 67
calculating, programmatically 68
content identifier 58
cryptocurrencies 58
data integrity verification 56, 57
digital signatures 57
Hash-based Message Authentication Codes (HMACs) 57

need for 56
network protocols 57
password verification 58
proof-of-work 58, 59
mini-CA
running 279
running, openssl ca subcommand used 280, 281
Modification Detection Code 2 (MDC-2) 66

N

national cipher 23
national cryptographic hash functions 65
National Institute of Standards and Technology (NIST) 20, 60
National Security Agency (NSA) 60
Nettle 6
network protocols 57
Network Security Services (NSS) 7
OpenSSL, comparing with 7
non-blocking sockets 253
non-repudiation 72
non-self-signed certificate
generating 150-153
NSS library 7
number used once (nonce) 18

O

OAEP padding 105
OCSP responder 213
OCSP stapling 213, 245
OCSP, using in C programs
OCSP callback, implementing 219-224
OCSP callback, registering 219
program, running 224, 225
offset codebook (OCB) mode 32
Online Certificate Status Protocol (OCSP) 140, 213, 245, 289
using 213
using, in C programs 218, 219
using, on command line 214-217
certificate revocation status, providing via 301-304
OpenBSD 9
OpenSSH 9, 63
OpenSSL 3-6
available asymmetric encryption 98, 99
comparing, with BoringSSL 10, 11
comparing, with Botan 7, 8
comparing, with GnuTLS 6, 7
comparing, with LibreSSL 9, 10
comparing, with lightweight TLS libraries 8, 9
comparing, with NSS 7
compiling 42-44
component purpose 36, 37
downloading 35
history 4, 5
installing 35
key derivation functions, overview 87
linking 42-44
URL 35, 140
used, for verifying TLS peer certificate 194
OpenSSL 3.0 5
features 5, 6
OpenSSL BIOs 174-176
openssl ca subcommand 280
used, for running mini-CA 280, 281
OpenSSL compatibility layers 4
openssl dgst subcommand 77
openssl enc file
decrypting, with AES programmatically 49, 50
decryption program, implementing 50, 51

decryption program, running 52
encrypting, with AES programmatically 44, 45
encryption program, implementing 45-48
encryption program, running 49
OpenSSL library
initializing 41, 42
uninitializing 41, 42
openssl mac subcommand 77, 78
OpenSSL operation implementation providers 5
operating system kernel 5
Operating System (OS) 244
opportunistic TLS 165
oracle 30
Organization Validation (OV) certificates 146
output feedback (OFB) mode 32
Output Key Material (OKM) 84

P

Padding Oracle On Downgraded Legacy Encryption (POODLE) 169
partial hash inversion 58
passphrase 84
password 84
symmetric encryption key, deriving on command line 87, 88
symmetric encryption key, deriving programmatically 88, 89
versus symmetric encryption key 84
Password-Based Key Derivation Function (PBKDF) 84-86
properties 85
password hashing 64
Perfect Forward Secrecy (PFS) 164
PGP 66
PKCS #12 container 296
PKCS #12 container files
client certificates, packaging into 226-228
plaintext 16
Poly1305 8
POWER8 62
Power ISA 62
prehash function 124
preimage attacks 59
pre-master secret 167
Pretty Good Privacy (PGP) 23, 63, 97
Privacy Enhanced Mail (PEM) 103, 139, 300
private key 72, 96
proof-of-work 58, 59
propagating CBC (PCBC) mode 32
provider 5
pseudorandom cipher digit stream 18
Pseudorandom Function (PRF) 76, 167
Pseudo-Random Number Generator (PRNG) 18
public key 96
Public Key Cryptography Standard number 7 padding (PKCS #7 padding) 33
disadvantage 34
Public Key Infrastructure (PKI) 98, 244
Public or Private Key (PKEY) 102
PureEdDSA 124
Python 7

Q

quantum computing 20, 102

R

Random Number Generator (RNG) 123
RC2 cipher 23

RC4 cipher
  reviewing 22
RC5 cipher 23
RC cipher family 24
Rijndael algorithm 21
RIPEMD-160 66
Rivest-Shamir-Adleman (RSA) algorithm 98
  reviewing 122
  using, to decrypt programmatically 115
  using, to encrypt and decrypt on command-line 105-107
root CA certificate 141
  generating 281-287
root CA config file
  reference link 285
rsa-decrypt program
  implementing 116
  running 117
rsa-encrypt program
  implementing 108-110
  running 110, 111
RSA key exchange 164
RSA keypair
  generating 102-105
RSA security 99-102

S

S-boxes 24
Scrypt algorithm 88
secret key 72
Secure Multipurpose Mail Extension (S/MIME) 63
Secure Sockets Layer (SSL) 4
security
  assessing, of cryptographic hash functions 59, 60
security bits 20
SEED algorithm 23
SEED cipher 24
selective forgery attack 72
self-signed certificate
  generating 147-150
Server Name Indication (SNI) 167, 268
session key 99
SHA-0 61
SHA-0 hash functions
  reviewing 63, 64
SHA-1 61, 64, 65
SHA-1 hash functions
  reviewing 63, 64
SHA-2 65, 66
SHA-2 family
  hash functions 60
  reviewing, of hash functions 60, 61
SHA-2 functions 60
SHA-3 65, 66
SHA3-256 66
SHA-3 family
  reviewing, of hash functions 61, 62
SHA3 family
  functions 61
SHA-256 59
SHA-512 61
SHAKE128 62
SHAKE256 62
Shang Mi 2 (SM2)
  reviewing 125
signatures
  subcommands, for signing and verification of 127-129
SM3 65
SM4 algorithm 23
SM4 cipher 24
Sophie Germain Counter Mode (SGCM) mode 32

Source Code Management (SCM) 58
source or sink BIOs 174
spam 59
SSH 66, 77
SSL 3.0 64
SSLeay library 4
SSL/TLS library 3
standard block padding 33
stream cipher 18
versus block cipher 17, 18
Streebog 65
Subject Alternative Names (SANs) 179
SVE/SVE2 62
symmetric cipher
national cipher 23
overview, by OpenSSL 17
RC cipher family 24
reviewing, by OpenSSL 23
symmetric cipher security 19
symmetric cryptographic algorithms 20
symmetric cryptography 4
symmetric encryption algorithm 4, 16, 17, 164
symmetric encryption key 84
deriving, from password on command line 87, 88
deriving, from password programmatically 88, 89
generating 34
versus password 84
Synthetic Initialization Vector (SIV) 32

T

TLS 66, 76
TLS 1.0 64
TLS 1.1 64
TLS certificate pinning 244, 245
cert_verify_callback() function, implementing 249-251
cons 246
pros 245
run_tls_client() function, modifying 247-249
tls-cert-pinning program, running 251-253
using 246
TLS client certificates
generating 225, 226
loading 236-240
program, running 232-234
requesting 228
response generation function, implementing 230-232
using 225
verifying 229, 230
verifying, on server side programmatically 228
TLS client connection
code inherited from tls-client program, modifying 235, 236
establishing, on command line 170-172
establishing, programmatically 176, 177
establishing, with client certificate programmatically 234, 235
program, running 240, 241
tls-client program, implementing 177-182
tls-client program, running 182
tls-client program
implementing 177-182
running 182
TLS connection 7, 194
TLS handshake 5, 166, 167
consequences 167, 168

TLS on non-blocking sockets
- run_tls_client() function, modifying 254-260
- tls-client-non-blocking program, running 260, 261
- using 254

TLS on non-standard sockets 262, 263
- run_tls_client() function, reimplementing 266-273
- service_bios() function, implementing 264-266
- tls-client-memory-bio program, running 273, 274
- using 263, 264

TLS peer certificate
- verifying, with OpenSSL 194

TLS protocol 164, 165
- history 168-170

TLS server connection
- accepting, on command line 173, 174
- accepting, programmatically 183, 184
- certificates, preparing 172, 173
- tls-server program, implementing 184-190
- tls-server program, running 190-192

tls-server program
- implementing 184-189
- running 190-192

TLS socket 5

TLS (Transport Layer Security) 4

TLS Working Group (TLS WG) 169

Transmission Control Protocol (TCP) 165

Transport Layer Security (TLS) 98, 243

Triple DES (3DES) 21

TrueCrypt 66

Trusted Third Parties 141

twisted Edwards curves 124

two-clause BSD license 8

U

universal forgery attack 72

User Datagram Protocol (UDP) 165

V

VeraCrypt 66

verify depth 195

W

web of trust 7, 98

Whirlpool 66

wolfSSL 8

X

x86/x86_64 62

X.509 63, 66

X.509 certificate 138-140
- fields 139

X.509 certificates 3, 4, 57, 164
- generating 144-146
- generating, stages 144

X.509 Public Key Infrastructure (PKI) 147

X509v3 extensions 146, 147, 301

x509-verify program
- implementing 155-158
- running 158

XOR (eXclusive OR) 18

Z

zero round-trip time (0-RTT) 170

Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

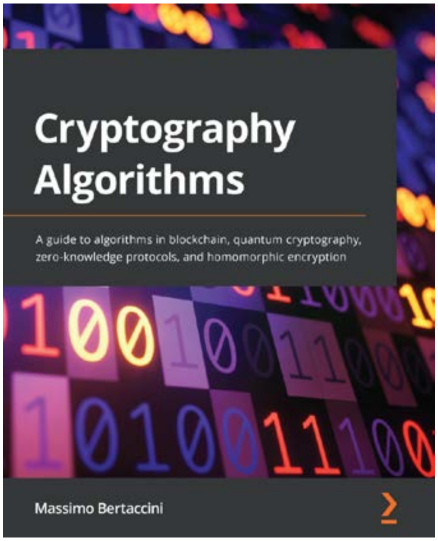
- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:

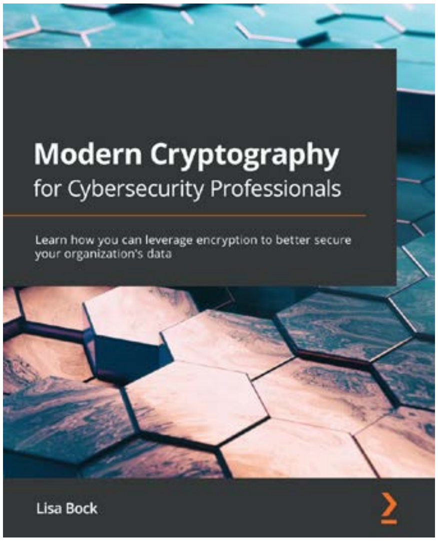

Cryptography Algorithms
A guide to algorithms in blockchain, quantum cryptography, zero-knowledge protocols, and homomorphic encryption
Massimo Bertaccini

Cryptography Algorithms

Massimo Bertaccini

ISBN: 978-1-78961-713-9

- Understand key cryptography concepts, algorithms, protocols, and standards
- Break some of the most popular cryptographic algorithms
- Build and implement algorithms efficiently
- Gain insights into new methods of attack on RSA and asymmetric encryption
- Explore new schemes and protocols for blockchain and cryptocurrency
- Discover pioneering quantum cryptography algorithms
- Perform attacks on zero-knowledge protocol and elliptic curves
- Explore new algorithms invented by the author in the field of asymmetric, zero-knowledge, and cryptocurrency


Modern Cryptography
for Cybersecurity Professionals
Learn how you can leverage encryption to better secure your organization's data
Lisa Bock

Modern Cryptography for Cybersecurity Professionals

Lisa Bock

ISBN: 978-1-83864-435-2

- Understand how network attacks can compromise data
- Review practical uses of cryptography over time
- Compare how symmetric and asymmetric encryption work
- Explore how a hash can ensure data integrity and authentication
- Understand the laws that govern the need to secure data
- Discover the practical applications of cryptographic techniques
- Find out how the PKI enables trust
- Get to grips with how data can be secured using a VPN

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit z and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Hi!

I am Alexei Khlebnikov, author of *Demystifying Cryptography with OpenSSL 3*. I really hope you enjoyed reading this book and found it useful for increasing your productivity and efficiency in securing your applications or networks with OpenSSL.

It would really help me (and other potential readers!) if you could leave a review on Amazon sharing your thoughts on *Demystifying Cryptography with OpenSSL 3*.

Go to the link below to leave your review:

`https://packt.link/r/1800560346`

Your review will help me to understand what's worked well in this book, and what could be improved upon for future editions, so it really is appreciated.

Best Wishes,

Alexei Khlebnikov